Obstacle Detection System in Dashcams using Convolutional Neural Networks

K Sathvik*, Karthik A Bhat, Mahanth M V., Nagavarshini*

Department of Information Science, NIE, Mysuru, India

DOI: https://doi.org/10.21467/proceedings.1.53
* Corresponding author email: sathvikkiran@yahoo.co.in

Abstract

Obstacle detection is an essential safety feature in all modern cars. Dashcams can be used to record traffic footage. The images obtained from dashcam have to be analyzed to detect the obstacles. Obstacles in the images need to be classified based on their distinctive properties. For a classification task, the object's feature has to be detected. It is not reliable to hard code to detect all features of any object as it will reduce the accuracy of prediction. A Neural Network is a better approach where it will determine the filters needed to classify the object into its respective class. Hence, a class of deep, feed-forward neural networks called convolutional neural networks has been used to analyze the imagery. In this paper, based on convolutional neural network, an efficient and accurate system to identify obstacles using the dash cam footage is being devised.

Index Terms- Artificial Intelligence (AI), Convolutional Neural Networks(CNN) , Red Green Blue(RGB), Inception, Convolutional Layer.

1 INTRODUCTION

Neural Network usually involves a large number of processors operating in parallel and arranged in tiers. The first tier receives the raw input information -- analogous to optic nerves in human visual processing. Each successive tier receives the output from the tier preceding it, rather than from the raw input -- in the same way neurons further from the optic nerve receive signals from those closer to it. The last tier produces the output of the system. Each processing node has its own small sphere of knowledge, including what it has seen and any rules it was originally programmed with or developed for itself. Neural networks are notable for being adaptive, which means they modify themselves as they learn from initial training and subsequent runs provide more information about the world. The most basic learning model is centered on weighting the input streams, which is how each node weights the importance of input from each of its predecessors. Inputs that contribute to getting right answers are weighted higher. A CNN is a class of deep, feed-forward artificial neural networks in machine learning used to analyze images. It follows the biological model of connectivity pattern between neurons in the animal visual cortex. It comprises of convolutional layers followed by one or more fully connected layers similar to a standard multilayer neural network. This is

 ISBN: 978-81-936820-0-5

achieved with local connections and tied weights followed by pooling which results in translation invariant features.

2 ARCHITECTURE

A CNN consists of a number of convolutional and subsampling layers optionally followed by fully connected layers. An m x m x r image is given as input to a convolutional layer where m is the height and width of the image and r is the number of channels, e.g. an RGB image has r=3. The convolutional layer will have k filters of size n x n x q where n is smaller than the dimension of the image and q can either be the same as the number of channels r or smaller. The size of the filters gives rise to the locally connected structure which are each convolved with the image to produce k feature maps of size m−n+1.

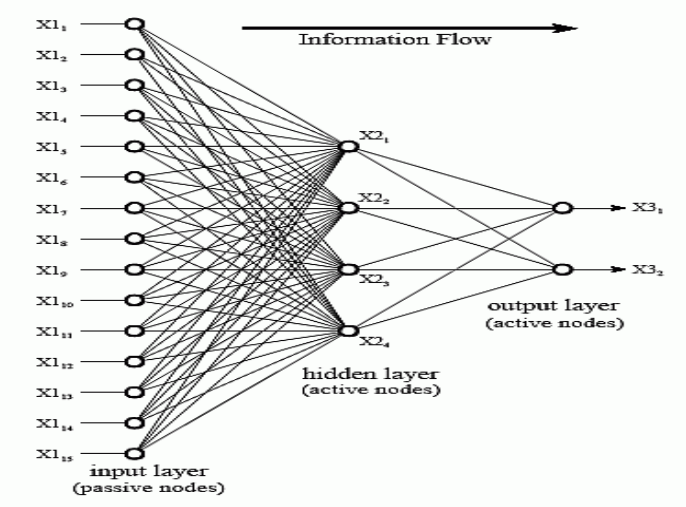


Fig-1: Neural Network architecture

The architecture of VGG network consists of 3*3 convolutional layers, 2*2 max pooling layers and fully connected layers at the end. It takes an input image of size 224*244*3(RGB image) to detect any one of 1000 images. It has a top-1 accuracy of 70%. AlexNet comprises of only eight layers; the first five are convolutional layers and the remaining three are fully connected layers. It has a top-1 accuracy of 57.1%. Among a number of neural network models developed by Google, Inception is preferred in the field. It uses the concept of a residual network with skip connections where the input is added to the output so that the model is forced to predict the residual rather than the target itself. This achieves 80.2% top-1 accuracy and 95.2% top-5 accuracy on the Imagenet dataset.

The deployment of VGG on most modest sized GPUs poses a problem because of huge computational requirements, both in terms of memory and time. Inception uses a bottleneck layer (1X1 convolutions) to help massive reduction of the computation requirement. The fully-connected layers at the end are replaced by a simple global average pooling which significantly reduces the total number of parameters in Inception. Thus, Inception is preferred in our experiments.

ISBN: 978-81-936820-0-5
Proceedings DOI: 10.21467/proceedings.1

Series: AIJR Proceedings

Inception-v2 to Inception-v3 results (single model)

Network	Top-1 Error	Top-5 Error	Cost Bn Ops
GoogLeNet [20]	29%	9.2%	1.5
BN-GoogLeNet	26.8%	-	**1.5**
BN-Inception [7]	25.2%	7.8	2.0
Inception-v2	23.4%	-	3.8
Inception-v2 RMSProp	23.1%	6.3	3.8
Inception-v2 Label Smoothing	22.8%	6.1	3.8
Inception-v2 Factorized 7×7	21.6%	5.8	4.8
Inception-v2 BN-auxiliary	**21.2%**	**5.6%**	4.8

- Each row's Inception-v2 model adds a feature with respect to the previous row's model
- The last line's model is referred to as the *Inception-v3* model

Fig-2: Comparison chart of inception-v3 and others

3 BACKPROPAGATION

Backpropagation is a supervised learning algorithm for training Multi-layered Neural Networks. In the beginning, while designing a Neural Network, we initialize weights with some random values. The initial values of the weights are never correct. Hence, the output of the model will be largely deviated from the actual desired output i.e. the error value is huge. To reduce the error, we re-train Inception using backpropagation. It calculates the gradient of the error function with respect to the neural network's weights.

4 DATASET

A driving car encounters various obstacle. The dashcam footage is used as the primary input for the network. The obstacles recorded by the dashcam are classified into different categories e.g. pedestrian, bicycle, dog. Inception is re-trained with different variants of individual categories to increase accuracy of prediction.

Fig-3: Dataset images for training

5 PROPOSED SYSTEM

Fig-4: Frame divided into patches

It is a tough challenge to correctly classify the given image into its actual category(class), because of varying lighting and camera angle when picture of the object is taken. The major objective is to recognize the object in front of the car via a dashboard camera into its respective category. This will help the driver and different research organizations such as Uber and Google's self-driving car to identify objects. In order to increase the accuracy of the prediction, Neural Network is used. The proposed system uses a trained convolutional neural network model (Inception) to analyze the obstacles from the image. Like any other classifier, the Neural Network Model has to be re-trained with images and its specific labels initially. After the re-training is complete, the model can successfully detect the obstacles in the images. The Neurons in hidden layer acts as feature extractor and SoftMax layer acts as classifier in the network. Since the proposed system sends an alert based on input image/frame, the driver need not check any other monitor while driving (it can be dashcam video or thermal cam video etc.), thus maintaining his focus on the road.

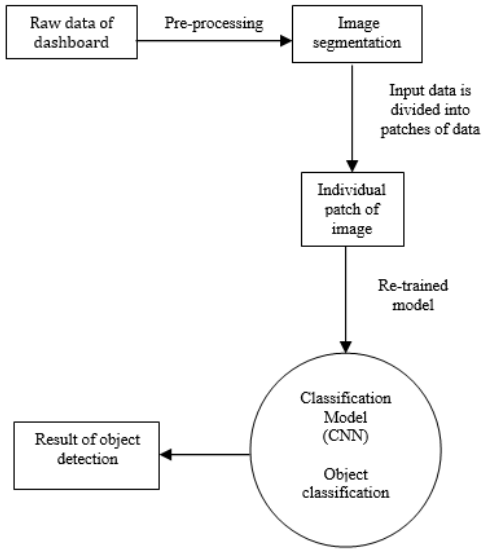
5.1 System architecture

A system architecture or systems architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.

The dashcam footage is taken as the input. The footage is broken down into individual frames. It is further divided into several patches. Each individual patch is then sent to Inception for object recognition. If the accuracy for the predicted category is greater than 70%, an alert message is sent as an output. Otherwise, ignored. Pre-image processing of patches has been done to obtain faster results.

ISBN: 978-81-936820-0-5
Proceedings DOI: 10.21467/proceedings.1

Series: AIJR Proceedings



5.2 Training

Typically, a neural network is initially trained or fed large amounts of data. Training consists of providing input and telling the network what the output should be. To build a network to identify the obstacles, initial training might be a series of pictures of pedestrians, bicycles, animals and so on. Each input is accompanied by the matching identification. Providing the answers allows the model to adjust its internal weightings to learn how to do its job better.

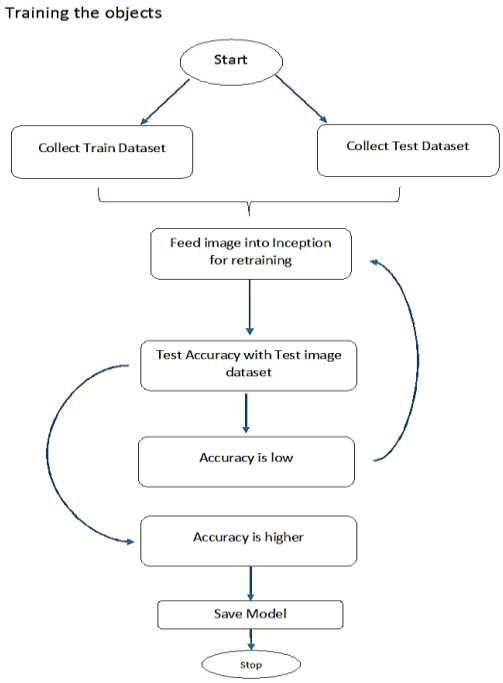


Fig-5: Flow diagram of training step

6 future work

The performance of the model may be further increased in the following manner:

- The number of patches that the frame is divided into could be made environment sensitive i.e. the frame will be divided into higher number of patches in city environments where the probability of obstacles is higher, while the number of patches may be reduced in highway environments.
- Motion estimation can be used for motion detection. In every frame, only the patches which detect motion need to analyzed. This makes computation quicker.
- Infrared cameras can be used for improving night vision.
- The proposed system can be used to perform Automatic emergency braking or autonomous emergency breaking (AEB).

7 CONCLUSIONS

The proposed system provides a warning system to the driver or the self-driving assistant. The system helps to avoid collisions and accidents while travelling. In future, Advancement to this system can be made by including auto breaking and auto steering by considering additional inputs along with dashcam footage.

References

[1] Xiaoling Xia, Cui Xu and Bing Nan, "Inception-v3 for Flower Classification",2017 Second International Conference on Image, Vision and Computing, DOI: 10.1109/ICIVC.2017.7984661

[2] Dennis Lui, "Deep neural networks changing the autonomous vehicle landscape", August 2017

[3] Hidenori Ide and Takio Kurita, "Improvement of learning for CNN with ReLu activation by sparse regularization", 2017 International Joint Conference on Neural Networks, DOI: 10.1109/IJCNN.2017.7966185

[4] W. Chen, J. T. Wilson, S. Tyree, K. Q. Weinberger, and Y. Chen. Compressing neural networks with the hashing trick. CoRR, abs/1504.04788, 2015.

[5] J. Hays and A. Efros. Large-Scale Image Geolocalization. In J. Choi and G. Friedland, editors, Multimodal Location Estimation of Videos and Images. Springer, 2014. 6, 7

[6] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio. Quantized neural networks: Training neural networks with low precision weights and activations. arXiv preprint arXiv:1609.07061, 2016.

[7] M. Jaderberg, A. Vedaldi, and A. Zisserman. Speeding up convolutional neural networks with low rank expansions. arXiv preprint arXiv:1405.3866, 2014.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages1097–1105, 2012.

[9] V. Lebedev, Y. Ganin, M. Rakhuba, I. Oseledets, and V. Lempitsky. Speeding-up convolutional neural networks using fine-tuned cp-decomposition. arXiv preprint arXiv:1412.6553, 2014.

[10] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi. Xnornet: Imagenet classification using binary convolutional neural networks. arXiv preprint arXiv:1603.05279, 2016.

[11] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems, 2015.

ISBN: 978-81-936820-0-5
Proceedings DOI: 10.21467/proceedings.1

Series: AIJR Proceedings